



Perry, R., Bull, DR., & Nix, AR. (1998). Pipelined DFE architectures using delayed coefficient adaptation. *IEEE Transactions on Circuits and Systems II - Analogue and Digital Signal Processing*, 45(7), 868 - 873. [7]. <https://doi.org/10.1109/82.700934>

Peer reviewed version

Link to published version (if available):
[10.1109/82.700934](https://doi.org/10.1109/82.700934)

[Link to publication record in Explore Bristol Research](#)
PDF-document

University of Bristol - Explore Bristol Research

General rights

This document is made available in accordance with publisher policies. Please cite only the published version using the reference above. Full terms of use are available:
<http://www.bristol.ac.uk/red/research-policy/pure/user-guides/ebr-terms/>

REFERENCES

- [1] P. B. Denyer and D. Renshaw, *VLSI Signal Processor—A Bit-Serial Approach*. Reading, MA: Addison-Wesley, 1985.
- [2] S. Y. Kung, *VLSI Array Processors*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [3] S. G. Smith *et al.*, "Techniques to increase the computational throughput of bit-serial architectures," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, ICASSP, Apr. 1987, pp. 543–546.
- [4] K. K. Parhi, "A systematic approach for design of digit-serial signal processing architectures," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 358–375, Apr. 1991.
- [5] R. Hatley and P. Corbett, "Digit-serial processing techniques," *IEEE Trans. Circuits and Systems*, vol. 37, pp. 707–719, June 1990.
- [6] M. D. Ercegovic, "On-line arithmetic: An overview," *SPIE*, vol. 495, pp. 86–93, 1984.
- [7] A. Aggoun, M. K. Ibrahim, and A. Ashur, "A novel cell architecture for high performance digit-serial computation," *Electron. Lett.*, vol. 29, no. 11, May 1993.
- [8] M. K. Ibrahim, "Radix-2" multiplier structures: A structured design methodology," *Inst. Elect. Eng. Proc.*, July 1993, vol. 140, Pt. E, no. 4, pp. 185–190.
- [9] A. E. Bashagha and M. K. Ibrahim, "Radix digit-serial pipelined divider/square-root architecture," *Inst. Elec. Eng. Proc. Comput. Digit. Tech.*, Nov. 1994, Vol. 141, no. 6, pp. 375–380.
- [10] J. V. McCanny *et al.*, "The use of data dependence graphs in the design of bit-level systolic arrays," *IEEE Trans. Acoust., Speech, and Signal Processing*, vol. 38, no. 5, pp. 787–793, May 1990.
- [11] G. Privat, "A novel class of serial-parallel redundant signed-digit multipliers," in *IEEE Int. Symposium on Circuit and Systems*, ISCAS, 1990, pp. 2116–2119.
- [12] M. C. Chen, "The generation of a class of multipliers: Synthesizing highly parallel algorithms in VLSI," *IEEE Trans. Comp.*, vol. 37, no. 3, Mar. 1988.
- [13] C. W. Wu and P. R. Cappello, "Block multipliers unify bit-level cellular multiplications," *Int. J. Comp.-Aided VLSI Design*, pp. 113–124, 1989.
- [14] K. Hwang, *Computer Arithmetic Principle, Architecture, and Design*. New York: Wiley, 1979.
- [15] C. R. Baugh and B. A. Wooley, "A two's complement parallel array multiplication algorithm," *IEEE Trans. Comp.*, vol. c-33, pp. 1045–1047, 1983.
- [16] J. V. McCanny *et al.*, "Optimized bit level systolic array for convolution," *Inst. Elec. Eng. Proc.*, Vol. 131, pt. F, no. 6, pp. 632–637, Oct. 1984.

Pipelined DFE Architectures Using Delayed Coefficient Adaptation

R. Perry, David R. Bull, and A. Nix

Abstract—In this paper the delayed least-mean-square (DLMS) algorithm is proposed for training a transversal filter-based decision feedback equalizer (DFE). Delays in the filter coefficient update process are used to pipeline the DFE, thereby increasing the throughput rate, for a given speed of hardware. The filter structures selected for the feedforward and feedback section of the DFE facilitate the use of a shared error signal, thereby reducing communication costs. The new resulting structure is highly modular and is very suitable for very large scale integration (VLSI) implementation. A pipelined form for the normalized least-mean-square algorithm (NMLS) is also obtained which removes the dependency of the convergence speed on the input signal power. The convergence and residual mean-square-error characteristics of the different pipelined filters are compared.

Index Terms—Adaptive filters, delayed least-mean-squares, equalization, pipelined.

I. INTRODUCTION

Although numerous adaptive equalizers have been reported in the literature, decision feedback equalization (DFE) remains a popular choice of equalization technique, especially for high-data-rate (e.g., HIPERLAN) applications, where alternatives, such as Viterbi equalizers, are precluded due to their complexity [1]. The use of long training sequences enables the selection of relatively simple gradient-based adaptive algorithms for equalizer training. Despite this concession, the realization of a high-sampling-rate DFE still remains challenging. This is due to the inherent sampling rate limitation of any adaptive filter associated with the generation and feedback of the error signal required for the filter coefficient updates. Exploitation of the natural parallelism in the least-mean-square (LMS) algorithm can help to reduce the algorithm iteration period. However, the regularity of the filtering structure is limited [2] and global broadcasting of the error signal to all coefficient update modules is still required. This has motivated the use of approximations to the LMS algorithm which sacrifice performance to increase the level of pipelining in the adaptation feedback loop. The delayed LMS (DLMS) algorithm is an example of an approximate form of the LMS algorithm, where the coefficient updates are delayed by an arbitrary number of sample periods [3].

In this brief a DFE architecture is derived, comprising a cascade of identical processing modules (PM's). The *order recursive filter* (ORF), described in [4], is used for the feedforward section but cannot be used for the feedback filter (FBF) because of the latency in the output. A transposed direct-form transversal filter (TF) together with a modified coefficient update is used to realize a pipelined FBF. By exploiting shared input parameters between the feedforward filter (FFF) and FBF stages, a new modular DFE structure is obtained (Section II-A), requiring minimal global communication.

Despite the attraction of its relative simplicity, a particular problem of implementing the LMS algorithm is the dependency of the algo-

Manuscript received August 1, 1996; revised April 1, 1997. This work was supported by Hewlett Packard Laboratories, Bristol, and by the U.K. Engineering and Physical Sciences Research Council (EPSRC). This paper was recommended by Associate Editor K. K. Parhi.

The authors are with the Centre for Communications Research, University of Bristol, Bristol BS8 1TR, U.K. (e-mail: Dave.Bull@bristol.ac.uk).

Publisher Item Identifier S 1057-7130(98)05061-7.

rithm stability and convergence speed on the power and correlation properties of the input data [5]. If a fixed step size is used, this must be kept sufficiently small to ensure stability under all expected operating conditions. A potential solution is to provide some degree of automatic gain control (AGC) by scaling the input data prior to equalization such that the step size \times power product is maintained within some predetermined design limits [6]. A practical means of implementing this is to use a data-dependent step size which is set to be inversely proportional to the sum of the square values of the current filter inputs. This approach results in the normalized LMS (NLMS) algorithm. Using the pipelining techniques described above, a pipelined form of the NLMS algorithm is also obtained in Section II. In Section III the performance of the pipelined DFE's are compared with a conventional realization of the LMS algorithm.

II. PIPELINED ADAPTIVE DFE'S USING DELAYED COEFFICIENT UPDATES

In this section it is shown how the introduction of a delay in the coefficient update process can be used to pipeline a transversal DFE using the basic LMS algorithm for training (Section II-A). In Section II-B a pipelined DFE structure adapted using the NLMS algorithm is developed. A comparison with other existing adaptive filter structures is presented in Section II-C.

A. DLMS Pipelined DFE

The complex form of the DLMS algorithm is given by (1) and (2)

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \beta e^*(n-D)\mathbf{X}(n-D) \quad (1)$$

$$e(n) = d(n) - \mathbf{W}^H(n-1)\mathbf{X}(n) \quad (2)$$

where $\mathbf{W}(n)$ is the vector of filter coefficients, $\mathbf{X}(n)$ is the vector of input data fed into the filter, β is the step size, and $e(n)$ is the error between the desired response $d(n)$ and the estimate of the desired response, produced by the equalizer at time n [7]. D is an arbitrary delay introduced in the gradient term $e^*(n-D)\mathbf{X}(n-D)$. For the case of a DFE, $\mathbf{W}(n)$ and $\mathbf{X}(n)$ are first partitioned into feedforward and feedback vectors and the delay D is set to L , the length of both the FFF's and FBF's (equal length filters are assumed without loss of generality). Equation (1) can be written in terms of these partitioned vectors as in (3)

$$[\mathbf{W}_f(n), \mathbf{W}_b(n)] = [\mathbf{W}_f(n-1), \mathbf{W}_b(n-1)] + \beta e^*(n-L)[\mathbf{X}_f(n-L), \mathbf{X}_b(n-L)] \quad (3)$$

where

$$\begin{aligned} \mathbf{X}_f(n) &= [x_f(n) \quad x_f(n-1) \quad \cdots \quad x_f(n-L+2) \\ &\quad \cdot x_f(n-L+1)]^T \\ \mathbf{X}_b(n) &= [x_b(n) \quad x_b(n-1) \quad \cdots \quad x_b(n-L+2) \\ &\quad \cdot x_b(n-L+1)]^T \\ \mathbf{W}_f(n) &= [w_f^0(n) \quad w_f^1(n) \quad \cdots \quad w_f^{L-2}(n) \quad w_f^{L-1}(n)]^T \\ \mathbf{W}_b(n) &= [w_b^0(n) \quad w_b^1(n) \quad \cdots \quad w_b^{L-2}(n) \quad w_b^{L-1}(n)]^T. \end{aligned}$$

Note that $\mathbf{X}_b(n)$ is the vector of previous training symbols, thus $x_b(n) = d(n-1)$. During decision-directed mode, $\mathbf{X}_b(n)$ will contain the previously detected symbols. $\mathbf{X}_f(n)$ is the vector of input samples to the FFF. The pipelining approach adopted here is to decompose the inner product computation in (2) into a cascaded series of pipelined stages, the output of the final stage being the value of the complete inner products of the FFF and FBF sections. The DFE output at time $L-1$, $y_{f,L-1}(n-L+1)$ is composed of two parts: the FFF output $y_{f,L-1}(n-L+1)$ and the FBF output $y_{b,L-1}(n-L+1)$

$$y_{f,L-1}(n-L+1) = y_{f,L-1}(n-L+1) + y_{b,L-1}(n-L+1).$$

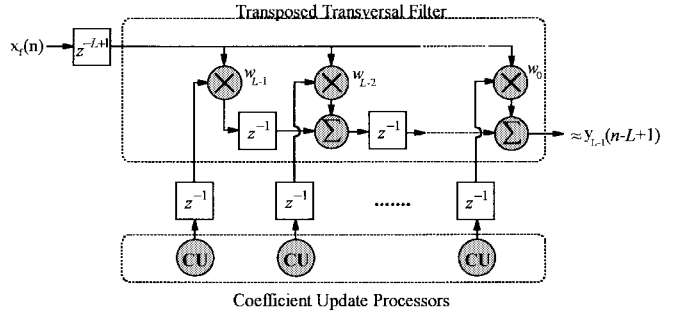


Fig. 1. Direct transposition of the adaptive FIR filter (TF1).

The contribution from the FFF and FBF's will be considered separately. Firstly, in a manner similar to [4], an output vector for the FFF output is defined as

$$\mathbf{Y}_f(n) = [y_{f,0}(n) \quad y_{f,1}(n-1) \quad \cdots \quad y_{f,L-2}(n-L+2) \quad y_{f,L-1}(n-L+1)]$$

where $y_{f,i}(n-i)$ represents the output of an i th-order inner product at time $n-i$, which is computed as

$$y_{f,i}(n-i) = \sum_{k=0}^i x_f(n-i-k)w_f^{k*}(n-i-1) \quad (4)$$

Equation (4) can be written order-recursively as

$$y_{f,i}(n-i) = y_{f,i-1}(n-i) + x_f(n-2i)w_f^{i*}(n-i-1) \quad (5)$$

from which it is recognized that $y_{f,i-1}(n-i)$ is the delayed output from the $(i-1)$ th-order filter [4]. It is clear from (5) how a pipelined ORF, for the FFF, may then be realized.

Pipelining the FBF is, however, more problematic. This is because the postcursor intersymbol interference (ISI) from the L previously detected symbols must be removed immediately from the FFF output at each iteration. For the recursion given in (5), the latency introduced using the ORF will prevent cancellation of the postcursor ISI distorting the equalizer output. Consequently, an alternative filtering structure is required for the FBF. An obvious choice, is a transposed direct-form TF, denoted TF1, shown in Fig. 1, which is inherently pipelined. Although transposition may be applied to a transversal filter with fixed coefficients while still preserving the same input-output characteristics, this is no longer true if the filter coefficients are time varying. Using TF1, the output of the FBF with the input delayed by i is given by

$$y_{b,L-1}(n-i) = \sum_{k=0}^{L-1} x_b(n-i-k)w_b^{k*}(n-1-k). \quad (6a)$$

From (6a), it is clear that the time indexes of the filter coefficients are skewed in time with respect to one another and thus this structure does not represent a strict realization of the DLMS algorithm. However, the time skew can be removed by delaying the coefficient updates as shown in the filter denoted TF2 in Fig. 2. The z^{-L+1} delay in this figure ensures the equivalence to the original ORF. This structure is not, however, used in its given form for the FBF, as the delays introduced in the coefficient updates can be distributed to yield a filter structure with improved pipelining.

For the TF2 structure, the filter output is described by a modified form of (6a), where an additional delay of $(L-1-k)$ sample periods is included in the k th coefficient time index. In this case $y_{b,L-1}(n-i)$ is given by

$$y_{b,L-1}(n-i) = \sum_{k=0}^{L-1} x_b(n-i-k)w_b^{k*}(n-1-k-(L-1-k))$$

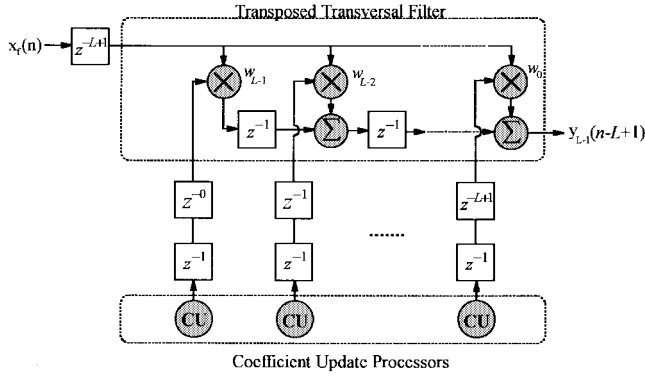


Fig. 2. Introduction of delays in the transposed structure (TF2).

and with time delay $i = L - 1$

$$\begin{aligned} y_{b,L-1}(n-L+1) &= \sum_{k=0}^{L-1} x_b(n-L+1-k)w_b^{k*}(n-L) \\ &= \mathbf{W}_b^H(n-L)\mathbf{X}_b(n-L+1). \end{aligned} \quad (6b)$$

In environments where the change in the channel ISI is small over D symbol periods, then $w_b^k(n-D) \cong w_b^k(n-k-1)k \leq D$. The output of the filter in Fig. 1 (TF1), following convergence, will then be similar to that from the filter in Fig. 2 (TF2). Consequently, both structures can be considered for use as the FBF. The DFE structure using TF1 will be referred to as the transposed direct-form DFE (TFDFE) and the DFE using a retimed version of TF2 will be referred to as the DLMS DFE. Using the form of (6b) for $y_{b,L-1}(n-L+1)$, a diagram showing the data flow for a (3,3) DLMS DFE is given in Fig. 3. The input to the FFF section enters from the left while the previous decision is input to *all* of the FBF sections simultaneously. The index for the FFF coefficients increases left to right, but for the FBF coefficients it decreases left to right (because of the transposition). The latency of the filter is $2L - 1$ sample periods. This is the time for the FFF to fill and for the estimate of the desired response to propagate along the filter structure.

Updates for the filter coefficients are now derived. For the FFF, the i th coefficient update is obtained from (3) as

$$w_f^i(n-i) = w_f^i(n-1-i) + \beta e^*(n-L-i)x_f(n-L-2i). \quad (7)$$

For the two alternative forms of $y_{b,L-1}(n-L+1)$, there are different coefficient updates. Using (6a) to compute $y_{b,L-1}(n-L+1)$ requires that $w_b^i(n)$ is obtained at each iteration. Using (3), this is given by

$$w_b^i(n) = w_b^i(n-1) + \beta e^*(n-L)x_b(n-L-i) \quad (8a)$$

Using (6b) to compute $y_{b,L-1}(n-L+1)$ requires the updated coefficients $w_b^i(n-(L-1-i))$. Using (3) gives

$$\begin{aligned} w_b^i(n-(L-1-i)) &= w_b^i(n-1-(L-1-i)) + \beta e^* \\ &\quad \cdot (n-2L+1+i)x_b(n-2L+1). \end{aligned} \quad (8b)$$

In (8a) the previous error term is broadcast to all of the FBF sections. In (8b) the same feedback data sample $x_b(n-2L+1)$ is broadcast to all of the FBF sections. Since the feedback data can be represented with a very short wordlength [1], broadcasting the feedback data sample is advantageous to reduce the global interconnection. It should also be noted that the reversed ordering of the filter coefficients in the FBF results in an identical error term in the update of the i th FBF coefficient in (8b) to that of the $(L-1-i)$ th FFF coefficient in (7). Thus, the error term can be shared between the update modules for the FFF and FBF coefficients. This is an attractive feature of

the TF2 structure compared to the TF1 structure since it reduces the communication costs in the DFE. This leads to the adoption of a modular processing section for the DLMS DFE structure as shown in Fig. 4, using the update in (8b). The step size β is shown multiplying the gradient estimates for the FFF's and FBF's in multipliers M2 and M5. In practice it is likely that β will be constrained to be a power-of-two value. Thus, the multipliers M2 and M5 can be replaced with hardwired or programmable shifts.

The estimate of the desired response, output from the final PM stage, is quantized to the nearest constellation point by a decision device, determined by the modulation format. This quantized output is used as the reference sequence during decision-directed mode and must therefore be computed within the same clock period as the computation of the equalizer output. Assuming the computation of the error is also completed within the same clock cycle, then the clock period will be determined as $t_m + 2t_a$, where t_m and t_a are the times required to perform a multiplication and addition, respectively. This assumes that multiplication by the step size β is reduced to a shift and that $t_s < t_a$, where t_s is the time required to perform a shift. By delaying the coefficient update by a further iteration period, the error computation can be eliminated from the critical path. The critical path will then be the longer of either the time to compute the DFE output and decision, in the final stage, or the time required to perform the coefficient update, i.e., $t_m + t_s + t_a$.

It is noted that since the DFE is trained using the DLMS algorithm, the convergence and stability analysis, previously developed in [3], is applicable here.

B. Pipelined Delayed NLMS DFE

To reduce the sensitivity of the LMS convergence speed on the input signal power, gain control has been suggested [6]. This can be realized by dividing the input data samples by an amount proportional to the estimated mean signal power, thereby adjusting the input signal power to within some predetermined range. However, the estimation of the received signal power, from the input data samples, imposes a delay before equalization may take place and can be unresponsive to rapid variations in the input signal statistics. For these reasons, the *NLMS algorithm* has been adopted which is able to achieve the same effect as AGC. In this algorithm the step size is adjusted in proportion to the inverse of the sum of the squares of the inputs to the adaptive filter [9]. By incorporating a delay in the coefficient update as before, the delayed NLMS algorithm (DNLMS) is obtained. The coefficient update for the DNLMS algorithm proposed here is given by

$$\mathbf{W}(n) = \mathbf{W}(n-1) + \beta(n-D)e^*(n-D)\mathbf{X}(n-D) \quad (9)$$

where $\beta(n)$ is a time-varying step size defined as

$$\beta(n) = \frac{\beta}{\mathbf{X}^H(n)\mathbf{X}(n)}. \quad (10)$$

The scaling factor β controls the speed of adjustment and is chosen in proportion to the length of the FFF's and FBF's. It will be shown below that $\beta(n)$ can be computed order-recursively in the same manner as the error signal. Note that during startup, when the FFF is not completely full of input data samples, the variable step size will be undetermined and, hence, must be initialized to some fixed value during this period. Since the computation of the error as used in (9) is the same as in (2), it can be performed in the same way as was described in Section II-A. The coefficient updates must, however, be modified to account for the time-varying step size. For the ORF, the feedforward coefficient updates are given by

$$\begin{aligned} w_f^i(n-i) &= w_f^i(n-1-i) + \beta(n-L-i) \\ &\quad \cdot e^*(n-L-i)x_f(n-L-2i). \end{aligned} \quad (12)$$

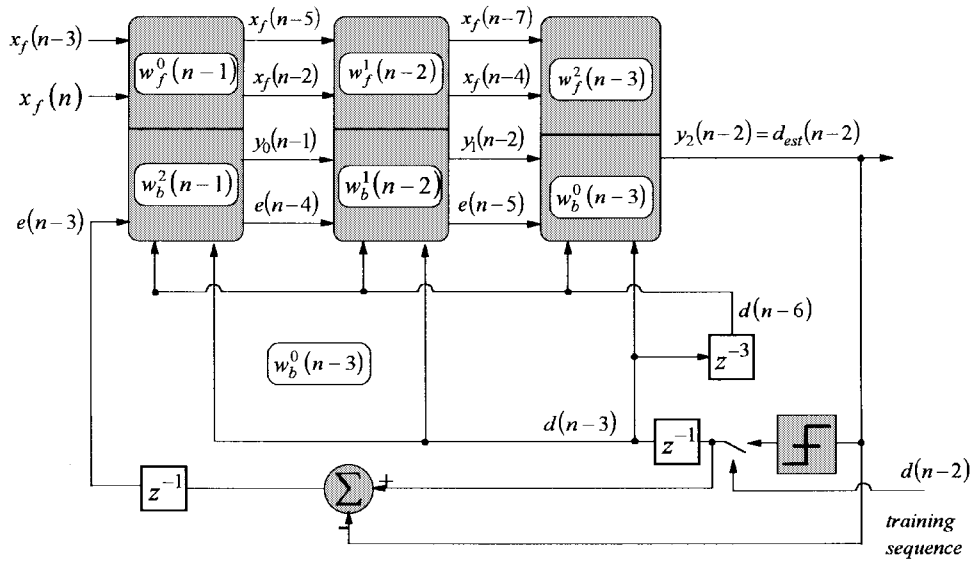


Fig. 3. Data flow in the pipelined (3,3) DLMS DFE.

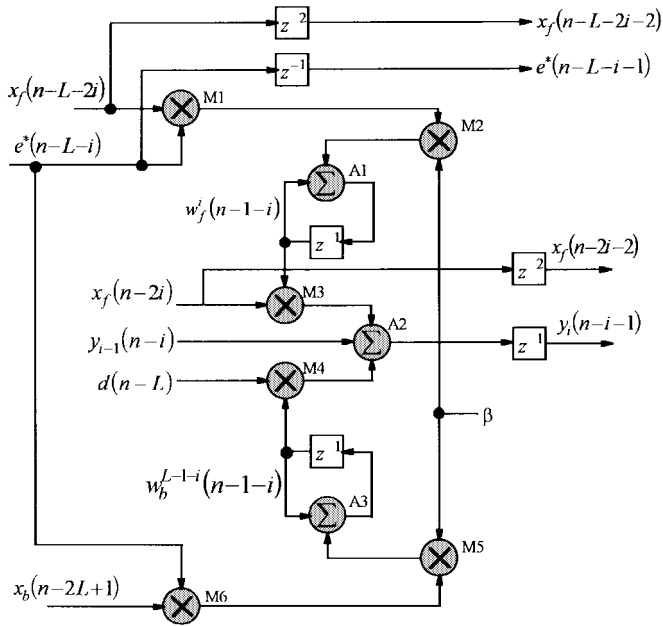


Fig. 4. A single PM for the pipelined DLMS DFE.

The computation of $\beta(n-L-i)$ requires the value of the inner product $\mathbf{X}^H(n-L-i)\mathbf{X}(n-L-i)$. Using the partitioned form of $\mathbf{X}(n)$ in (3), the inner product $\mathbf{X}^H(n)\mathbf{X}(n)$ can be expressed as

$$\begin{aligned}\mathbf{X}^H(n)\mathbf{X}(n) &= \mathbf{X}_f^H(n)\mathbf{X}_f(n) + \mathbf{X}_b^H(n)\mathbf{X}_b(n) \\ &= Z_{L-1}^f(n) + Z_{L-1}^b(n)\end{aligned}\quad (13)$$

where

$$Z_{L-1}^f(n) = \sum_{k=0}^M |x_f(n-k)|^2 \quad Z_{L-1}^b(n) = \sum_{k=0}^M |x_b(n-k)|^2.$$

It should be noted that when the FBF is *full*, in the case of a nonamplitude modulated training sequence, $Z_{L-1}^b(n)$ is a constant equal to $L|d(n)|^2$. In such a case, the contribution to the inner product in (12) from the FBF inputs does not explicitly require computation. It thus remains to compute an update for $Z_{L-1}^f(n)$. From (11), the

update of $w_f^i(n-i)$ requires $Z_{L-1}^f(n-L-i)$. Now, from (12)

$$Z_{L-1}^f(n-i) = \sum_{k=0}^i |x_f(n-i-k)|^2 \quad (14)$$

which, in the same manner as for the data estimate calculation in (5), can be computed order recursively as

$$Z_{L-1}^f(n-i) = Z_{L-1}^f(n-i-1) + |x_f(n-i)|^2. \quad (15)$$

This computation can thus be carried out in parallel with the other computations in the existing PM. The computation of $Z_{L-1}^f(n-i)$ can be integrated in each PM, with the value of $Z_{L-1}^f(n-L+1)$ becoming available in the last PM, from which the modified step size $\beta(n-L+1)$ is computed.

It has been implicitly assumed in the above derivation that a variable step size is used in the coefficient updates of both the FFF and FBF coefficients. It is readily shown, however, that only a *variable* step size for the FFF is necessary to remove the dependency of the convergence speed on the received signal power. In [6] the training sequence (and hence the values of the feedback data stream) was also scaled by the same factor as the input data. However, since the feedback data inputs are normally selected from a limited alphabet of symbols, this can be readily exploited to reduce the complexity of the inner product computation. Scaling the training sequence by a variable factor would preclude this. If a fixed step size is used for the FBF updates, the term $Z_{L-1}^b(n)$ may be omitted from (12).

Implementing a variable step size for the FFF will severely reduce the throughput rate because of the need for a division operation. In addition, there is an increase in the interconnect cost associated with the transmission of an additional parameter (the variable step size) between each of the PM's. To avoid this, the product of the step size and estimation error should be computed once at the output of the final DFE stage. The DFE throughput can be maintained by including an additional M delays in the coefficient updates, sufficient to pipeline the division and multiplication operations. The equalizer output is then approximated as

$$y_{L-1}(n-L+1) \cong \mathbf{W}^H(n-L-M)\mathbf{X}(n-L). \quad (16)$$

Provided M is reasonably small, the effect on convergence will not be significant and will eliminate the error and step-size computations from the critical path. The critical path will now be in the coefficient update with associated computational delay of $2t_m + t_a$.

C. Comparison with Other Pipelined Architectures

In a previous publication by the authors [8] and in a recent publication [9] the transposed form of a transversal filter was used in a realization of the DLMS algorithm. In [9], a factor two slowdown was employed to introduce extra delays for pipelining, in addition to delaying the coefficient adaptation. Hardware utilization efficiency was maintained by folding two operations onto each processing element. This reduced the number of multiplier elements, which dominate the total chip area. It was argued that the structure was more area efficient, for a given sampling rate, than the architectures in [4] and [10] (the structure in [10] was the first systolic array proposed for the DLMS algorithm) and, thus, also the structures used in this paper. However, the comparison made in [9] did not take into consideration some straightforward simplifications of the structure in [4] which enable a significant reduction in the sampling period. Firstly, the critical path in the filter proposed in [4] was wrongly assessed in [9], since the error computation time was combined with the time required to perform the coefficient update. As described in Section II-A, the error computation should be included in the critical path associated with the formation of the DFE output. Secondly, the multiplication of the step size with the error term can be easily factored out of the critical path, as noted in Section II-B. Finally, the provision of a multiplication for the step-size operand is excessive and, in practice, a power-of-two value is generally sufficient [10]. Using these simple modifications, the critical path delay in the adaptive filters used to construct the DFE in Section II-A is $t_m + t_a$. The adaptive filters proposed in [9] have minimum sampling periods of $2(t_m + t_a)$ and, therefore, do not provide the same performance as the filters described here. This is an obvious consequence of using fewer arithmetic elements. Circuit folding [12] may be applied to the DFE structure in Fig. 4, allowing an approximate factor of two reduction in the number of arithmetic elements, at the expense of a similar reduction in the throughput. The resulting filters provide virtually identical performance to those described in [9], i.e., the same hardware requirements for the same throughput rate. It is noted that an efficient DFE structure using a fractionally spaced FFF can be realized by the application of circuit slowdown and folding to the DFE structure in Fig. 4.

Other techniques to increase the throughput of adaptive filters have been proposed. In [13], very high throughput rate architectures were proposed, utilizing fine-grain pipelining of the arithmetic elements. Pipelining has been used here to restructure the DFE in order to ease implementation as well as increase throughput. In addition, for mobile applications (e.g., [1]), the throughput rate of the proposed pipelined DFE is adequate, while avoiding excessive increases in the convergence time (Section III) which would reduce the bandwidth efficiency. It is also noted that fine-grain pipelining of a DFE is not possible, since the DFE output must be available at the end of each iteration in order to cancel the effects of postcursor ISI. Although in [13] look-ahead was applied as a means of overcoming this inherent throughput limitation, the resulting algorithm was nonlinear and has to be considerably approximated in order to simplify implementation. In [14], multiple DFE's were proposed in order to increase the throughput rate by processing different sections of a transmitted frame of data. By the insertion of known symbols to correctly initialize the DFE's, a speedup factor equal to the number of DFE's is achievable. While this technique can, in principle, provide unlimited speedup, the hardware overhead is likely to limit the number of DFE's actually used. The method can, however, be used in conjunction with pipelining as proposed in this paper.

III. EQUALIZER PERFORMANCE COMPARISON

The simulated performance of the pipelined DFE's are compared using the channel models proposed in [5] and [15]. Channels 1 and

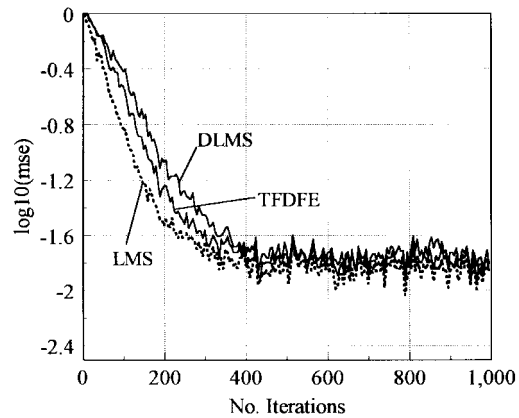


Fig. 5. Convergence comparison between the LMS and DLMS algorithms over channel 1.

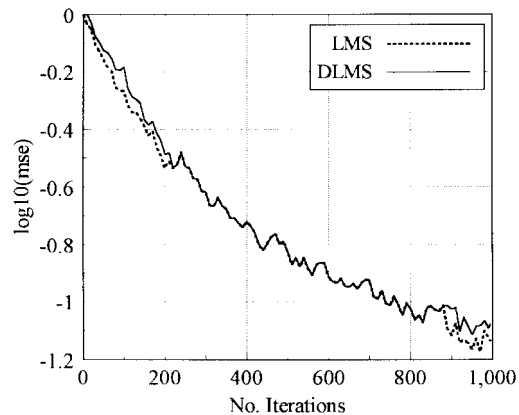


Fig. 6. Convergence comparison between the LMS, DLMS, and TFD FE algorithms over channel 3.

2 are given by channels "a" and "b" from [5], respectively; channel 3 is from [15]. The performance of the pipelined DFE realizing the DLMS algorithm for training is compared with the conventional LMS algorithm and the TFD FE structure discussed in Section II-A. In the convergence curves, the log of the mean-square-error, normalized with respect to the squared value of the training data, is plotted. For clarity, not all points on the convergence curves are shown. In the simulation a quadrature phase-shift keying (QPSK) modulated signal is used with root-raised-cosine filtering at the receiver and transmitter. Additive noise was added with the E_b/N_0 figure for the system fixed at 20 dB. The convergence curves were generated by averaging over 150 independent experiments.

In Fig. 5 the convergence of the (12,12) DFE structures are compared over channel 1. The DFE has been overspecified in this experiment in order to exaggerate the differences in the convergence behavior. The behavior of the DLMS and TFD FE algorithms are noticeably different. The convergence speed of the LMS algorithm is superior for the given step size 0.015. As expected, the convergence speed of the TFD FE lies between that of the DLMS and LMS algorithms. This is due to the additional delay in the filter coefficient updates used in the DLMS algorithm.

Channel 3 is a much more severe test than channel 1 since it introduces a null in the frequency response. It was found for equalization of channel 3 that a reasonable compromise between performance and convergence speed could be obtained using a (10,10) DFE structure with $\beta = 0.01$. The convergence behavior of the DLMS and LMS algorithms is compared in Fig. 6; the TFD FE

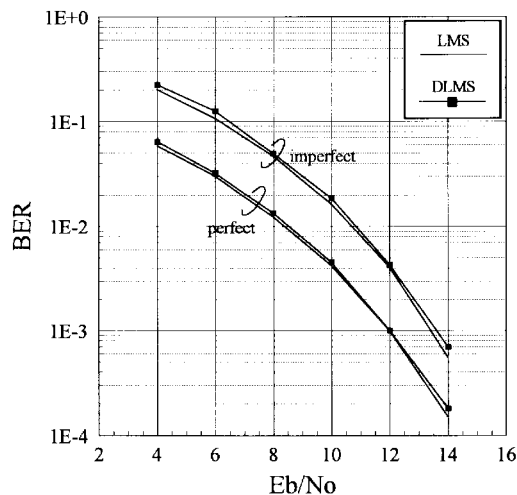


Fig. 7. BER comparison of the DLMS and LMS algorithms over channel 2 ($\beta = 0.01$).

performance was similar to the DLMS algorithm. In these cases the reduced step size and more severe channel distortion resulted in very little performance variation between the three algorithms.

In Fig. 7, the performance of the LMS and DLMS algorithms are compared in terms of the bit-error rate (BER) over channel 2 for a (6,6) DFE (500 training symbols were allocated). Perfect decision feedback and imperfect decision feedback were compared. The performance loss in this case is only marginal, since the delay in adaptation is not very large. This is expected since, for both algorithms, convergence was reliably achieved within the 500-symbol training period.

IV. CONCLUSION

In this paper pipelined transversal filter-based DFE's employing the DLMS training algorithm have been described. An order-recursive DFE structure was developed which allows a DFE of arbitrary length to be constructed by cascading a series of identical processing modules. Alternative filtering structures were chosen for the FFF's and FBF's in order to minimize the global communication. The performance of the new DFE's were compared using simulated channels to introduce ISI and were found to be only marginally inferior to those for the conventional DFE. However, the pipelined DFE's more than double the throughput rate of conventional structures and are very suitable for VLSI implementation. A pipelined version of the DNLMS algorithm was also proposed for a DFE, which removes the dependency of the convergence speed on the input signal power.

ACKNOWLEDGMENT

The authors would like to thank the other members at the Centre for Communications, University of Bristol, and the reviewers of this paper for their helpful suggestions and for pointing out some of the references.

REFERENCES

- [1] A. Nix, M. Li, J. Marvill, T. Wilkinson, I. Johnson, and S. Barton, "Modulation and equalization considerations for high performance radio LAN's (HIPERLAN)," in *Proc. PIMRC*, vol. 3, The Hague, The Netherlands, Sept. 1994, pp. 964–968.

- [2] S. P. Smith and H. C. Torng, "A fast inner product processor based on equal alignments," *J. Parallel Distrib. Comput.*, vol. 2, no. 4, pp. 376–390, 1985.
- [3] G. Long, F. Ling, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 1397–1405, Sept. 1989.
- [4] M. D. Meyer and D. P. Agrawal, "A high sampling rate delayed LMS filter architecture," *IEEE Trans. Signal Processing*, vol. 40, pp. 727–729, Nov. 1993.
- [5] J. G. Proakis, *Digital Communications*, 2nd ed. New York: Macmillan, 1989.
- [6] N. J. Bershad, "Analysis of the normalized LMS algorithm with Gaussian inputs," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 793–806, Aug. 1986.
- [7] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.
- [8] R. Perry, D. Bull, and A. Nix, "An adaptive DFE for high data rate applications," in *IEEE Proc. Vehicular Technology Conf. (VTC 1996)*, vol. 2, Atlanta, GA, Apr. 1996, pp. 686–690.
- [9] J. Thomas, "Pipelined systolic architectures for DLMS adaptive filtering," *J. VLSI Signal Processing*, vol. 12, pp. 223–246, June 1996.
- [10] H. Herzberg, R. Haimi-Cohen, and Y. Be'ery, "A systolic array realization of an LMS adaptive filter and the effects of delayed adaptation," *IEEE Trans. Signal Processing*, vol. 40, pp. 2977–2803, Nov. 1992.
- [11] H. Samueli, B. Daneshrad, B. C. Wang, and H. T. Nicholas, "A 64-tap CMOS echo canceller/decision feedback equalizer for 2B1Q HDSD transceivers," *IEEE J. Select Areas Commun.*, vol. 9, pp. 839–847, Aug. 1991.
- [12] K. K. Parhi, C.-Y. Wang, and A. P. Brown, "Synthesis of control circuits in folded pipelined DSP architectures," *IEEE J. Solid-State Circuits*, vol. 27, pp. 29–43, 1992.
- [13] N. R. Shanbhag and K. K. Parhi, *Pipelined Adaptive Digital Filters*. Norwell, MA: Kluwer, 1994.
- [14] A. Gatherer and T. H.-Y. Meng, "A robust adaptive parallel DFE using extended LMS," *IEEE Trans. Signal Processing*, vol. 41, pp. 1000–1005, Feb. 1993.
- [15] A. P. Clark and S. F. Hau, "Adaptive adjustment of receiver for distorted digital signals," *Proc. Inst. Elect. Eng.*, vol. 131, no. 5, pp. 526–536, Aug. 1984.

Intermodulation Noise Related to THD in Dynamic Nonlinear Wide-Band Amplifiers

Henrik Sjöland and Sven Mattisson

Abstract—In this brief it is shown that the power of the intermodulation noise of a wide-band amplifier with dynamic nonlinearities can be estimated by the total harmonic distortion (THD) with a sinusoid input signal of appropriate amplitude and frequency. The THD is, as opposed to the intermodulation noise, easy to measure and use as a design parameter. This brief is an extension of our paper [1], which treats static nonlinearities.

Index Terms—Distortion, intermodulation, wide-band amplifiers.

I. INTRODUCTION

In [1] it was shown that the intermodulation noise power due to a static nonlinearity can be estimated by a total harmonic distortion

Manuscript received September 5, 1996; revised July 7, 1997. This paper was recommended by Associate Editor V. Porra.

The authors are with the Department of Applied Electronics, Lund University, S-22100 Lund, Sweden (e-mail: hsd@tde.lth.se).

Publisher Item Identifier S 1057-7130(98)05062-9.